Extending QuickTime's Object VR Interactivity

Stewart Crawford

BioGraphix, LLC && Visible Productions, LLC; Fort Collins, CO USA
http://www.cs.colostate.edu/~sgcraw

Figure 1 : A series of images used to create a rotating skull.

Abstract

Image-based 3D visualizations are required when frame rendering times are long, when photographic image sets rather than mathematical models of objects are available, or when model complexity does not support real-time rendering. The Object VR model of QuickTime (QT) is a common framework for such visualizations. While these have proved useful in the biomedical community, the QTVR framework has some limitations, presented and addressed in this paper. This work provides solutions to the issues of smoothly going beyond two degrees of control and allowing models to freely tumble end-over-end; an improved method of hot spot labelling and interaction is also presented. These solutions are all being actively implemented in our current generation of biomedical models.

Keywords: QuickTime, object VR, QTVR, interactive images

CR categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – Virtual reality; I.3.3 [Computer Graphics]: Picture/Image Generation – Viewing algorithms

1. Introduction

Image-based interactive 3D visualizations, though less glorious and less studied than polygonal models rendered and texture mapped in real time, still have a useful lives. Image-based visualizations are still viable in cases where there is as yet no mathematical model of an object but a sequence of still photographs can be obtained, or when the complexity of the model dictates that it can't be rendered sufficiently quickly to support interactive frame rates.

The Object VR model embodied in QuickTime's QTVR suite of tools and players is perhaps the most common framework for these visualizations. The biomedical community has developed a large set of these Object VRs for university teaching and patient education [Nieder 2002]. We have utilized image-based Object VRs since our detailed polygonal models, when combined to show many anatomical systems, have polygon counts in the hundred-thousands with rendering times measured in minutes.

With user interaction controlled by the mouse, horizontal and vertical mouse motion provide two natural degrees of freedom to control the model viewing. In the biomedical community, we have used these two degrees of freedom to make either (1) fully rotatable models or (2) dissectable models with limited rotation. But we would like to have models both fully rotatable and dissectable, which imply more than two degrees of control. Another limitation of QTVRs is that they are inconsistent with a our usual experience of handling on object, since the QTVRs is limited in its ability to tumble freely, end-over-end, in all directions. In this paper, we provide solutions to these problems, and provide an improved alternative to QTVR's hot spot methodology which is used to identify and act upon specified regions in the imagery.

2. QuickTime Object VR Basics

QuickTime has two distinct flavors of VR in its QTVR toolkit: Panoramic and Object VR. The user's world model of a Panoramic VR is that they are standing in one place, viewing the world as they turn around. The user's world model of an Object VR is that they are holding an object and rotating it around in their hands observing it. This work addresses only the handling of Object VR imagery.

The simplest object model is a sequence of images taken around an object, as shown here in Figure 1. When these are frames of a movie and put in a loop, the skull would appear to be rotating around its vertical axis. To add user interaction, the user can be presented with any one image initially and then asked to drag the cursor across the image. Dragging the cursor to the right advances the movie frame-by-frame to the right, wrapping around appropriately at either end of the image sequence. The user thus appears to be rotating the object clockwise or counter-clockwise by interacting through the mouse with the image sequence. This is the most basic user-controlled Object Movie or Object VR, also referred to as "interactive images".

To allow the user to more freely rotate the object, additional

strips of images are used, with the object rotated about its vertical axis and the camera inclined at an azimuth angle. In Figure 2 (on the color plates), each strip represents a change of 20 azimuth degrees from its neighbors.

This represents the QTVR Object model. We call this a *global rotatable model*, to distinguish it from the more primitive single axis *rotatable model*. Note that while the left-most images wrap smoothly around to the right-most images on each row, the top images do not smoothly wrap around to the bottom in each column. Thus continued upward mouse motion, when positioned on the top row, has no effect.

For smoother transitions, the standard recommendation is to have increments of 10 degrees in both equatorial and azimuth angles, which results in an array of 19 rows of 36 images each, or 684 total images [Apple 2000]. On the QuickTime movie timeline, the rows are laid out linearly, adjacent to one another; horizontal mouse movements over an image move in the timeline frame-to-frame, while vertical mouse movements jump to the same relative frame in the appropriately adjacent group.

Object VRs have filled a niche in the biomedical education community [Gutmann 2000; Trelelease et al. 2000]. One innovation that receives use in this community is called a "dissectable VR". In this style of VR, the rows represent an object at various levels of dissection. For a dissectable model of the head, the first row shows the skin, the second row would show the bones and muscles, the third would show the brain, and additional rows would show succeedingly deeper structures within the brain. In this way, as the user moves the cursor vertically in an image, the user sees the progressive peeling away of structures, simulating a dissecting of the subject. A collection of both global and dissectable human anatomical models has been published on the Merk Medicus web site [Merk 2003]. The QuickTime VR Anatomical Resource web site [Nieder 2002] carries links to a further variety of anatomical object VR models.

Conceptually, nothing in this model requires the use of QuickTime; there just happen to be tools available in QuickTime toolkits that aid in assembling these representations. We have implemented this interaction on a DVD, using the up/down/left/right keys of a standard DVD controller to run the model. A straightforward java applet could implement this same user experience: it would read a super-image that essentially looks like Figure 2 (color plates), with all the images, full scale, aggregated into one image; it would display only one sub-image through a window; and as the user moves the cursor horizontally or vertically, the aggregate super-image is simply shifted under the viewer's window.

3. Limitations of QuickTime

We generated a series of these VRs for biomedical clients during the late 1990s. During the course of this work, several limitations were encountered:

* QuickTime worked well across both Mac and Windows platforms, but troubles arose across generations of Quicktime. We had models generated with Quicktime versions 2 through 5, and later players wouldn't run earlier generation models, which forced us to continually reload various QuickTime libraries.

* Mouse motion in 2D gave two-dimensions of control in the VRs. With horizontal cursor motion controlling rotation about a vertical axis, vertical cursor motion could be used to control either moving through dissection layers **or** rotation about horizontal axes. But now we wanted both, VRs that are at the same time globally rotatable and dissectable.

* The image strips of Figure 2 (color plates) wrap around horizontally, but not vertically. The user is thus manipulating a model which rotates smoothly, but when rotating end-over-end, it gets "stuck" at the top and bottom and won't tumble any further. This is at variance with our day-to-day tactile manipulation experience with physical objects: we can tumble the object over, seeing it right-side up, then on end, then up-side down, then on (other) end, then right-side up, then on end, ...

In addressing the versioning problem, we looked at rewriting the VR movie controller for web delivery in another way. We considered the applet mentioned earlier, but requiring novice users to load a JVM on their machine was judged both risky and as onerous as loading specific QuickTime versions. Director/Shockwave was a possibility. But Flash was the rising star, with a compact and soon-to-be ubiquitous player in all the web browsers. The basic Flash 5 controller solved our first problem, and our models still run in later players. As a side benefit, more platforms are covered, since Flash players are available across the unix/linux world, which has sparse QuickTime support. Now with a VR movie controller under our development, we could start addressing other VR issues.

4. End-Over-End Tumbling

The image set of Figure 2 (color plates) fully captures the skull at all aspects. Why can't we tumble the model and see it upside down? Starting with the skull face, if we tumbled it to see the bottom and continued tumbling, we'd soon see the back of the skull, upside down. Figure 3 shows the top half of the image matrix. The solution to the end-over-end tumble is to observe that, upon arriving at the top image in column three, the image sequence continues moving down column nine, as long as the images in column nine are rotated 180 degrees.

There is one issue: the viewer may notice the column switch depending on the object lighting and any image background. If the lighting setup and background are rotationally symmetric about the camera axis, there will be no visual effect if the images are rotated.

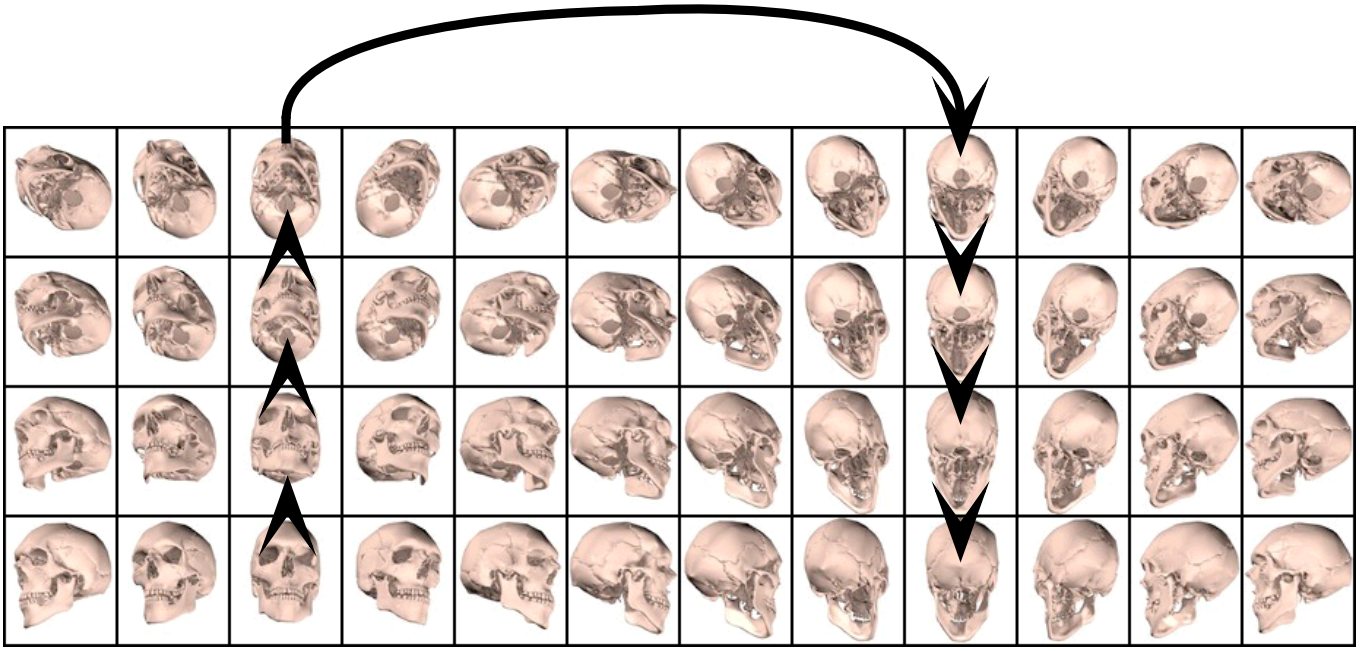

Figure 3 : The natural sequence of images, after moving up column three, continues down column nine, if column nine's images are rotated 180 degrees. A demoonstrataion SWF was submitted with the auxillary material.

5. Dissectable & Globally Rotatable VRs

With an end-over-end, globally rotatable VR now implemented as a Flash movie clip, dissectability can be added by layering movie clips on top of one another. Virtual dissections can now be achieved by swapping in new movie clips that are synchronized to the previous clip. Further flexibility can be achieved by layering the movie clips one on top of the other, synchronizing them with one another, and then smoothly varying the alpha transparency of the movie clips on top. The transparency of each clip can be varied independently creating a variety of effects, such as a ghost layer of skin overlaying translucent muscles overlaying almost solid organs with the trace view of the skeleton under it all.

With all this imagery, these models can grow quite large. A 440 x 440 sized VR, with 108 images per layer, five layers deep, with each image an alpha-channel PNG at 4 bytes-per-pixel, requires in 345.6 Mb of raw image data. While this image data will be compressed when stored as an SWF file, it all is brought into play when the Flash player runs the VR. This can slow down and potentially cause trouble with the Flash player, which was not likely to have been designed with this much throughput as a consideration.

6. Hot-spot Labelling

The ability to designate specific image areas of a QTVR (for both panoramic and object VRs) allows an image to be labelled or linked. These specially designated image areas are called *hot spots*. The QTVR interface allows the programming of various actions when the cursor rolls over a hot spot or when the user clicks the cursor within the boundary of a hot spot. Hot spots are delineated on an off-camera image that is mapped to the image within the user's view. Figure 4 (color plates) show an example of an image and its hot spot definitions.

Each of the named bones, ligaments, and muscles in the image on the left has a corresponding area painted a solid, unique color in the hot spot image on the right. These hot spot images are color-mapped, 8-bit images [Chen 1995], and thus limited to identifying at most 256 areas in their corresponding image.

In this example, the user has placed the cursor (in the left image) over a particular muscle, the semimembranosus muscle. The corresponding location in the hot spot image is blue. The color map index for this blue points into a table of information about the structure; in this application, the name of the muscle is retrieved and displayed to the user. The information associated with the colormap indices could represent almost anything: it could link the user to different VRs, it could generate a quiz for that structure, it could be an audio clip; it could contain physical therapy information. The basic idea is that the solid color in the hot spot image determines the region over which a specific action would occur.

This model of interaction doesn't directly port to a Flash implementation, because it has no primitives to query an image as to what color is at a particular location. Buttons with rollover states, though, can be put to similar effects. Typically, buttons on web pages are thought of as simple rectangular shapes. Flash buttons can have any shape, and they need not be contiguous. Thus, a set of buttons can be defined, one button corresponding to each of the uniquely colored areas in the hot spot image. Figure 5 (color plates) shows the muscle identified earlier. Note in this case, the rollover state was set to a semi-transparent white, which

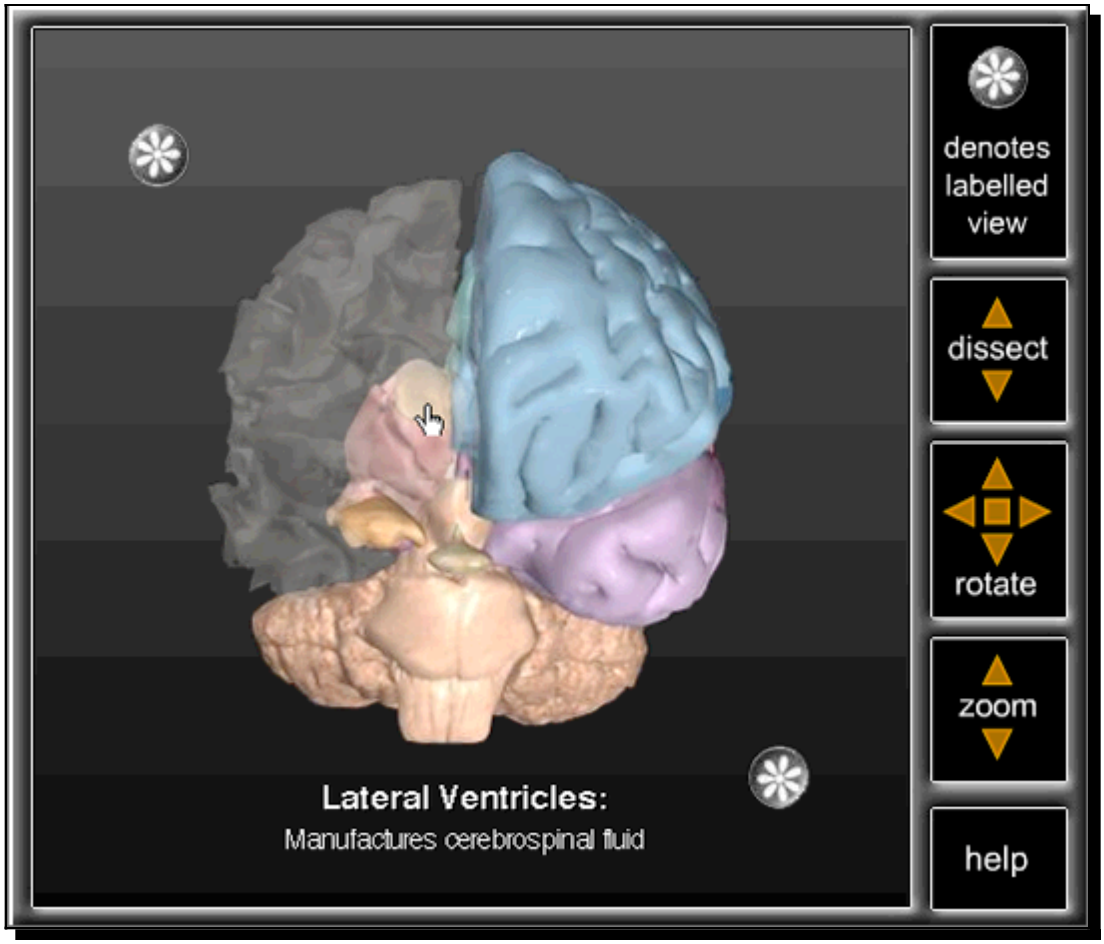

Figure 6 : A labelled, fully rotatable, five-layer dissectable visualization of the brain.

highlights the full extent of the muscle. Note the extension of the semimembranosus muscle on its lower portion – it is not obvious from the original image that the two muscle parts are related. This illustrates a shortcoming of the original, off-screen colormapping methodology: the cursor location can be mapped to a specific color map index, but it would be difficult to map back from the colormap to the original image and highlight all locations with identical properties.

This solution is compact in space (since the off-screen colormapped images are not needed) but expensive in processing, because now the player must search each of many irregularly shaped buttons deciding whether the cursor is within its region.

7. Discussion

This work provides solutions to the issues of allowing biomedical object visualizations to be simultaneously both fully rotatable and dissectable, and allowing the visualizations to freely tumble end-over-end. An improved method of general region labelling and interaction is also presented. These solutions are all being actively implemented in our current generation of biomedical models. The solutions presented here were implemented with Flash's actionscript, though they would be straightforward to implement in other languages. This has provided the next increment of additional functionality associated with Object VR models in general.

References

Apple Computer, Inc. 2000. *QuickTime for the Web*. Morgan Kaufman.

Chen, S.E. 1995. "QuickTime VR – An Image-Based Approach to Virtual Environment Navigation", In Proceedings of ACM SIGGRAPH 1995, ACM Press / ACM SIGGRAPH, New York. R.Cook, Ed. Computer Graphics Proceedings, Annual Conference Series, ACM, 29-38.

Merck & Co., Inc. 2003. "MerckMedicus : Your Key to the Medical Internet". http://www.merckmedicus.com/pp/us/hcp/frame_emedtool.jsp

Nieder, G.L. 2002. "QuickTime VR Anatomical Resources", http://www.anatomy.wright.edu/qtvr/.

Trelease, R.B., Nieder, G.L., Dørup, J., and Hansen, M.S, 2000. Going Virtual With QuickTime VR: New Methods and Standardized Tools for Interactive Dynamic Visualization of Anatomic Structures. In *The Anatomical Record (New Anat)* 261: 64-77.

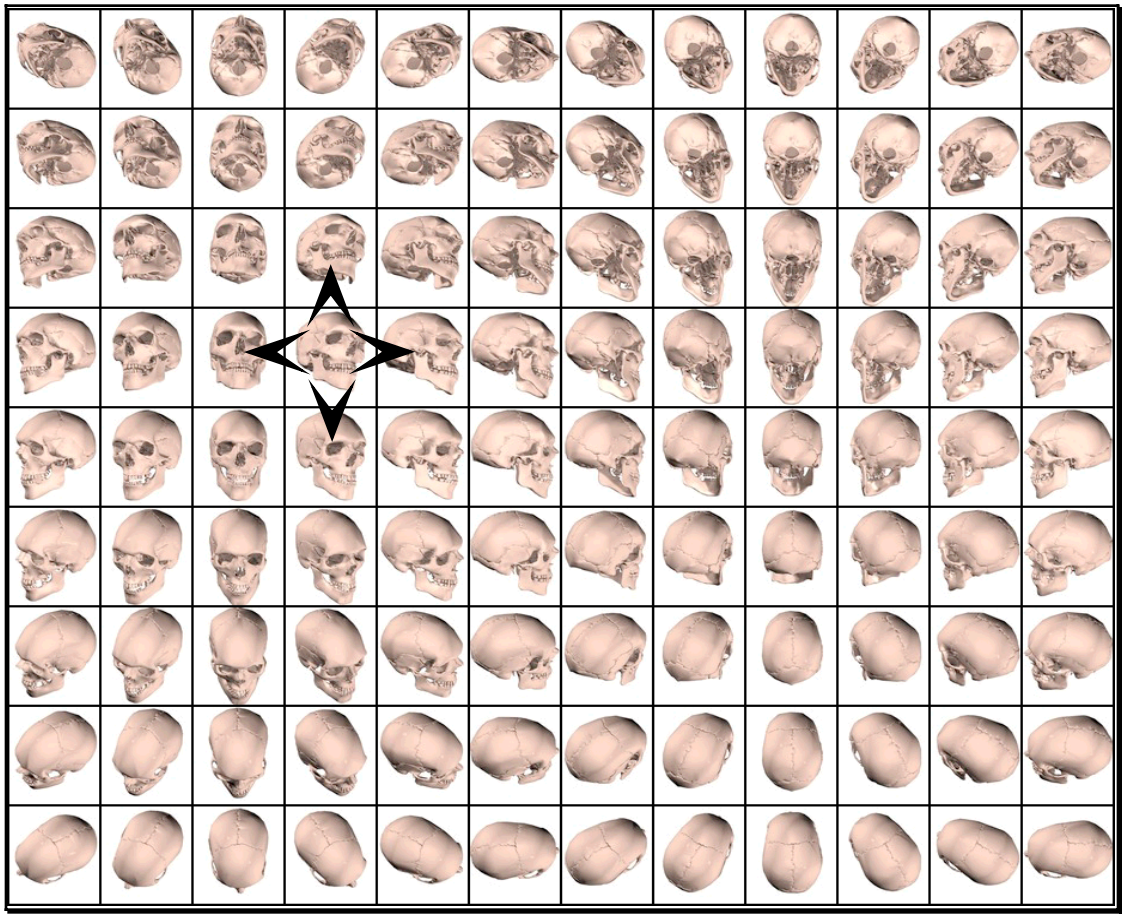

Figure 2: The image set used to make a fully rotatable Object VR of the skull. If the image in the fourth row, fourth column were currently displayed, the left arrow indicates which image would be displayed next corrsponding to a motion of the cursor. Though not shown, diagonal motions are also allowed.

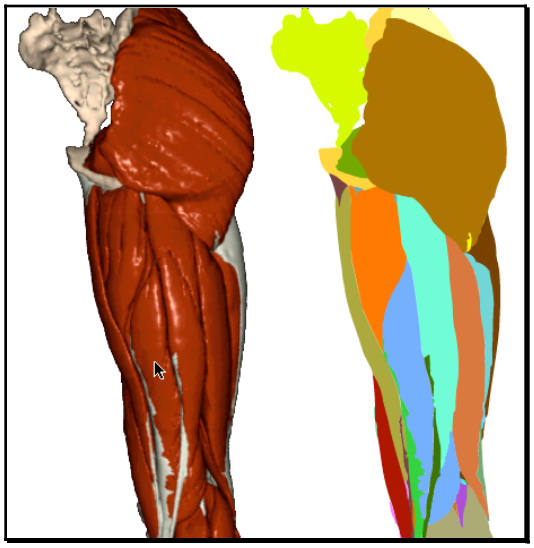

Figure 4: An image of the leg's muscles (left) and its corresponding hot spot map (right). Each unique hot spot color designates a particular structure in the original image.

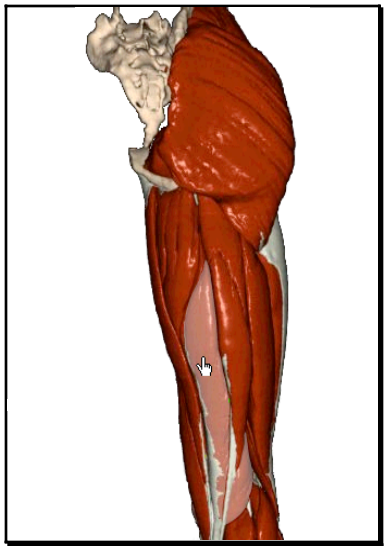

Figure 5: The roll-over state of an irregularly shaped button defines the muscle's full extent.